

# Dynamic Hybrid Forecasting Models for Drug Consumption Prediction in Hospital Pharmacies

Yuxin Fan\* and Siye Wu†

\*School of Engineering and Applied Science, University of Pennsylvania, Canada, Toronto

Email: yuxinfan@alumni.upenn.edu

†Simon Business School, University of Rochester, Canada, Toronto

Email: april.siyewu@hotmail.com

**Abstract**—Accurate and efficient drug consumption forecasting is crucial for hospital pharmacies to avoid overstocking, minimize wastage, and ensure continuous patient care. This study proposes a hybrid forecasting framework integrating XGBoost, Prophet, and SARIMAX to improve monthly consumption predictions at the drug-manufacturer level. Through rolling-window forecasting and advanced feature engineering, the proposed approach addresses challenges such as seasonality, trend shifts, and sparse data. Experimental results demonstrate significant improvements in prediction accuracy and robustness across diverse drug consumption scenarios.

**Index Terms**—Drug Consumption Forecasting, Hospital Pharmacies, Forecasting Models, XGBoost, Prophet, SARIMAX

## I. INTRODUCTION

Drug consumption forecasting plays a critical role in hospital pharmacy inventory management. Accurate predictions enable hospital pharmacies to ensure drug availability while minimizing costs associated with overstocking or stockouts. As highlighted by Koala et al. [1], forecasting drug consumption is particularly challenging due to numerous influencing factors, such as sociodemographic characteristics, morbidity patterns, drug price index, and seasonal factors like disease outbreaks or policy changes. This study aims to develop a hybrid forecasting framework tailored to address these challenges by integrating XGBoost, Prophet, and SARIMAX with rolling-window forecasting and advanced feature engineering.

In previous studies, various forecasting techniques have been proposed to address one or more of these complexities. Taylor and Letham [2] introduce Prophet, a scalable forecasting method designed for large-scale applications, focusing on capturing general trends and seasonality with changepoint detection. However, Prophet is limited in handling external variables and often requires careful tuning to avoid overfitting to local patterns. Machine learning techniques, such as XGBoost proposed by Chen and Guestrin [3], effectively model non-linear relationships but struggle with sequential dependencies in time-series data.

Furthermore, comparative and hybrid studies like those by Ferreira et al. [4] and Meng et al. [5] highlight that while LSTM improves non-linear pattern recognition compared to ARIMA and Prophet excels in handling sparse data, both face challenges with computational complexity and adaptability. Xu et al. [6] demonstrate a hybrid approach combining linear

regression and LSTM networks, which improves temporal forecasting but struggles with capturing complex, non-linear interactions across diverse datasets. Siddiqui et al. [7] explore hybrid models like ARIMA-Holt's Winter, which improve upon traditional statistical methods but rely heavily on fixed architectures, limiting adaptability in dynamic environments. Rathipriya et al. [8] utilize hybrid neural networks to address temporal patterns but face challenges with data sparsity and computational efficiency.

In this study, to address the above limitations, our proposed framework introduces rolling-window forecasting to adjust predictions for non-stationary data and shifting trends dynamically, along with advanced feature engineering that enriches the data with lag features and trend indicators, capturing both short-term dependencies and long-term dynamics. These techniques are seamlessly integrated with hybrid complementary models: XGBoost, which models non-linear relationships and feature interactions; SARIMAX, which incorporates external variables and seasonality; and Prophet, which handles trend decomposition and missing data. This synergy provides a robust and versatile solution to the unique challenges of drug consumption forecasting, making the framework applicable across various drug types and manufacturers.

The remainder of this paper is organized as follows: Section II details the methodology, including data preprocessing and model design. Section III presents experimental results and discussions, highlighting the advantages of the proposed framework. Finally, Section IV concludes the study and outlines future research directions.

## II. METHODOLOGY

### A. Hybrid Model Framework

The proposed hybrid forecasting framework integrates the complementary strengths of Prophet, SARIMAX, and XGBoost to address the diverse challenges in drug consumption prediction. Single-model approaches often struggle with generalization across datasets characterized by varying seasonality, non-linear dependencies, and sparsity. By leveraging the unique capabilities of each model, the hybrid framework provides a robust solution tailored to the specific characteristics of individual drug-manufacturer combinations.

Drug consumption data exhibits significant variability, with some combinations showing consistent trends while others

experience volatility or irregular demand. Models like Prophet excel at capturing stable long-term trends and seasonal patterns, making them suitable for drugs with high and regular demand. On the other hand, SARIMAX is better equipped to handle datasets influenced by external variables or strong short-term dynamics, as it integrates advanced feature engineering and external factor modeling. For scenarios with complex non-linear relationships or sparse observations, XGBoost proves advantageous due to its flexibility in handling feature interactions and non-linear dependencies.

To optimize predictions, the framework dynamically selects the most appropriate model for each drug-manufacturer combination. Performance metrics such as  $R^2$  and symmetric mean absolute percentage error (SMAPE) are employed to evaluate each model's effectiveness. A high  $R^2$  score indicates that the model captures most of the variance in the data, while a low SMAPE value reflects accuracy in representing relative changes. This data-driven model selection process ensures that the framework adapts to heterogeneous consumption patterns, maximizing prediction accuracy and robustness.

By dynamically assigning models based on their strengths, the hybrid framework efficiently handles diverse scenarios. Prophet is ideal for stable and predictable datasets, SARIMAX excels in addressing external factors and seasonality, while XGBoost captures non-linear and sparse patterns. This synergy of models enhances the overall robustness of the forecasting system, providing a scalable and versatile solution for hospital pharmacies managing a wide range of drug consumption trends.

### B. Model Design and Roles

The hybrid framework integrates Prophet, SARIMAX, and XGBoost, each addressing distinct aspects of drug consumption prediction. Prophet excels at decomposing time-series data into trends, seasonality, and special events, making it suitable for datasets with consistent patterns. SARIMAX incorporates external variables and advanced feature engineering to model short-term dependencies and seasonality, effectively handling data influenced by external factors. XGBoost captures non-linear relationships and complex feature interactions, providing robustness for datasets with irregular or sparse observations. By combining these models, the framework dynamically adapts to diverse data characteristics, ensuring accurate and robust predictions across a wide range of forecasting scenarios.

1) **SARIMAX Model:** SARIMAX extends the traditional ARIMA model by incorporating external (exogenous) variables, making it well-suited for time-series data with seasonality, trends, and contextual factors. The model is defined as:

$$y_t = \phi(B)\theta(B)^{-1}(c + \mathbf{X}_t\beta + \epsilon_t), \quad (1)$$

where  $y_t$  represents the target variable (e.g., monthly drug consumption),  $\mathbf{X}_t$  denotes the exogenous variables, and  $\epsilon_t$  is the white noise error term.

To enhance prediction accuracy, SARIMAX utilizes advanced feature engineering to incorporate exogenous variables, including:

- **Lagged values:** Capture delayed effects of past consumption:

$$\text{lag}_k = y_{t-k}, \quad k \in \{1, 3, 6, 12\}. \quad (2)$$

- **Rolling statistics:** Represent short-term trends and variability:

$$\text{Rolling Mean}_k = \frac{1}{k} \sum_{i=1}^k y_{t-i}, \quad (3)$$

$$\text{Rolling Std}_k = \sqrt{\frac{1}{k} \sum_{i=1}^k (y_{t-i} - \text{Mean})^2}. \quad (4)$$

- **Seasonality:** Encode monthly seasonality using trigonometric functions:

$$\text{Month\_sin} = \sin\left(\frac{2\pi \cdot \text{Month}}{12}\right), \quad (5)$$

$$\text{Month\_cos} = \cos\left(\frac{2\pi \cdot \text{Month}}{12}\right). \quad (6)$$

- **Exponential weighted moving average (EWMA):** Smoothing recent observations:

$$\text{EWMA}_\alpha = \alpha y_t + (1 - \alpha) \cdot \text{EWMA}_{\alpha, t-1}. \quad (7)$$

- **Percentage change and trend strength:** Capture relative variations and long-term dynamics:

$$\text{Pet\_Change}_k = \frac{y_t - y_{t-k}}{y_{t-k}}, \quad (8)$$

$$\text{Trend Strength} = \frac{1}{k} \sum_{i=1}^k |y_{t-i} - y_{t-i-1}|. \quad (9)$$

These engineered features enable SARIMAX to capture both short-term dependencies and long-term trends, significantly enhancing its ability to model complex temporal patterns in drug consumption data.

2) **XGBoost Model:** XGBoost is a gradient boosting framework that constructs an ensemble of decision trees to predict the target variable  $y_t$ . It is particularly effective for time-series forecasting involving non-linear relationships and sparse data. The model predicts  $y_t$  through an additive function:

$$\hat{y}_t = F(x_t) = \sum_{k=1}^K f_k(x_t), \quad f_k \in \mathcal{F}, \quad (10)$$

where  $\hat{y}_t$  is the predicted value,  $x_t$  represents input features,  $f_k$  denotes the  $k$ -th decision tree, and  $\mathcal{F}$  is the space of decision trees.

**Feature Engineering:** Temporal dependencies are captured using lagged values ( $y_{t-1}, y_{t-2}, \dots$ ), rolling statistics (e.g., moving averages and standard deviations over 3, 6, and 12 periods), and seasonal patterns encoded via trigonometric functions ( $\sin, \cos$ ). Interaction terms between lagged values and seasonal indicators further enhance the model's ability to capture complex dependencies.

**Rolling-Window Training:** The model employs a rolling-window approach to dynamically adapt to evolving trends. At each iteration, the training dataset is updated with the most recent observations, and the model is retrained to make

predictions for the next time step, ensuring adaptability and minimizing overfitting.

**Hyperparameter Optimization:** To maximize performance, a grid search tunes parameters such as:

- *Number of estimators:* Determines ensemble size.
- *Learning rate:* Controls the contribution of each tree.
- *Maximum tree depth:* Prevents overfitting by limiting tree complexity.
- *Subsample and column sampling ratios:* Enhance generalization by regulating data and feature subsets.

By modeling non-linear interactions and addressing irregular patterns, XGBoost complements statistical models like SARIMAX. Its integration with rolling-window training and advanced feature engineering ensures robust and adaptive predictions, making it a key component of the hybrid forecasting framework.

3) *Prophet Model:* Prophet is a time-series forecasting model developed by Facebook, designed to decompose data into trend, seasonality, and holiday components. It is particularly effective in handling missing values, outliers, and irregular patterns, making it suitable for dynamic drug consumption data. The model predicts the target variable  $y_t$  as:

$$y_t = g(t) + s(t) + h(t) + \epsilon_t, \quad (11)$$

where  $g(t)$  represents the long-term trend,  $s(t)$  captures seasonal patterns using Fourier series,  $h(t)$  accounts for holiday effects, and  $\epsilon_t$  is the white noise error term. This decomposition enhances interpretability while maintaining predictive accuracy.

**Hyperparameter Optimization:** Key hyperparameters are optimized via grid search:

- *seasonality\_mode:* Models seasonal effects as additive or multiplicative.
- *changepoint\_prior\_scale:* Controls flexibility for abrupt trend changes.
- *seasonality\_prior\_scale:* Adjusts the weight of seasonal components.

A higher *changepoint\_prior\_scale* enables the model to adapt to frequent structural changes, while lower values favor smoother transitions.

**Rolling-Window Forecasting:** The model employs a rolling-window strategy, retraining on the most recent data at each iteration. This ensures adaptability to emerging trends while mitigating the impact of outdated patterns, making it robust for datasets with irregular or sparse observations.

By combining decomposition capabilities, grid search optimization, and rolling-window forecasting, Prophet provides accurate and interpretable predictions, complementing the other components of the hybrid framework.

### C. Dynamic Rolling-Window Forecasting

Time-series data often exhibit non-stationarity, where trends, seasonality, and noise evolve over time. Static forecasting approaches relying on fixed historical data may struggle to capture these dynamics, leading to suboptimal performance. To address this, a dynamic rolling-window forecasting strategy

is employed, enabling models to prioritize recent information and adapt to structural changes.

At each time step  $t$ , the training dataset is updated to include the most recent observations while discarding older data beyond the defined window size ( $W$ ). Formally, the training dataset at  $t$  is defined as:

$$\mathcal{D}_t = \{(y_\tau, \mathbf{X}_\tau) \mid \tau \in [t - W, t - 1]\},$$

where  $\mathcal{D}_t$  is the training data,  $y_\tau$  denotes the target variable, and  $\mathbf{X}_\tau$  represents feature vectors. After training on  $\mathcal{D}_t$ , predictions are generated for the next time step ( $t + 1$ ).

The rolling-window approach effectively captures short-term dynamics by prioritizing recent patterns while mitigating the influence of outdated data. It is particularly advantageous for handling structural changes such as shifts in trends or seasonality. The choice of window size ( $W$ ) is critical: larger windows incorporate long-term patterns, while smaller windows emphasize recent changes. This study evaluates multiple window sizes and selects the optimal configuration based on metrics such as root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE).

Integrating rolling-window forecasting into the hybrid framework enhances adaptability:

- **SARIMAX:** Dynamically recalibrates coefficients to model short-term dependencies and external influences.
- **XGBoost:** Refines decision trees with updated feature interactions, capturing evolving non-linear relationships.
- **Prophet:** Updates its trend decomposition to align with the latest data.

This iterative strategy ensures the models remain responsive and resilient to changes, achieving superior performance across diverse drug consumption scenarios by continuously adapting to the most recent dynamics.

### D. Implementation and Optimization

The implementation of the proposed forecasting framework emphasizes modularity, scalability, and computational efficiency. Each model is designed to operate within a unified pipeline that supports dynamic updates, enabling seamless integration of rolling-window forecasting and hyperparameter optimization.

The forecasting process is organized into three primary stages: data preprocessing, model training, and evaluation. In the preprocessing stage, raw data is transformed into feature-rich datasets through advanced feature engineering, ensuring compatibility with the unique requirements of each model. The training stage employs dynamic rolling-window strategies, allowing each model to adapt to the most recent data while leveraging grid search to optimize key hyperparameters. Finally, the evaluation stage utilizes metrics such as root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE) to assess forecasting performance across diverse drug-manufacturer combinations.

To ensure computational efficiency, the framework incorporates parallel processing and model-specific optimizations. For example, XGBoost utilizes multi-threading capabilities to

accelerate decision tree construction, while Prophet leverages Fourier series approximations to efficiently model seasonal components. The SARIMAX implementation is enhanced by efficient matrix operations for parameter estimation, particularly when handling large datasets with multiple external variables.

The framework's modular design allows for seamless adaptation to new data sources and forecasting scenarios. Detailed implementation details, including pseudocode and algorithmic workflows, are provided in Appendix A to facilitate reproducibility and scalability for future applications.

### E. Dynamic Feature Engineering

Feature engineering is crucial for enhancing model performance. Besides the exogenous variables described above, outlier detection is a key preprocessing step.

1) *Outlier Detection and Handling*: Outliers can distort predictions and degrade model performance. This study employs two approaches for outlier detection:

- **Z-score Method**: The Z-score for each data point  $y_i$  is calculated as:

$$Z_i = \frac{y_i - \mu}{\sigma}, \quad (12)$$

where  $\mu$  is the mean and  $\sigma$  is the standard deviation of the dataset. Data points with  $|Z_i| > 3$  are identified as outliers.

- **Interquartile Range (IQR) Method**: The IQR is defined as:

$$\text{IQR} = Q_3 - Q_1, \quad (13)$$

where  $Q_1$  and  $Q_3$  are the 25th and 75th percentiles, respectively. A data point  $y_i$  is considered an outlier if:

$$y_i < Q_1 - 1.5 \cdot \text{IQR} \quad \text{or} \quad y_i > Q_3 + 1.5 \cdot \text{IQR}. \quad (14)$$

To mitigate the effect of outliers, values exceeding these thresholds are capped at the 5th and 95th percentiles of the data distribution:

$$y_i = \begin{cases} \text{Percentiles}, & \text{if } y_i < \text{Percentiles} \\ \text{Percentile}_{95}, & \text{if } y_i > \text{Percentile}_{95}. \end{cases} \quad (15)$$

2) *Rolling-Window Forecasting*: To adapt to dynamic consumption patterns, a rolling-window mechanism is implemented. At each prediction step  $t$ , the model is trained using historical data  $\{y_1, y_2, \dots, y_t\}$ , and the prediction for  $t + 1$  is made. The window then updates to include the latest observation, ensuring that the model adapts to recent trends.

## III. EXPERIMENTS

### A. Experimental Setup

The dataset consists of monthly drug consumption records, including:

- Drug Name
- Manufacturer
- Monthly Consumption

Models are evaluated using the following metrics:

#### 1) Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}. \quad (16)$$

#### 2) Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|. \quad (17)$$

#### 3) Symmetric Mean Absolute Percentage Error (SMAPE):

$$\text{SMAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2} \times 100\%. \quad (18)$$

### B. Results

The performance of the models is summarized in Table I.

## IV. DISCUSSION AND FUTURE WORK

### A. Model Comparison and Hybrid Framework

While each model has unique strengths and limitations, integrating their complementary capabilities within a hybrid framework provides significant advantages:

- **XGBoost** handles nonlinear relationships effectively.
- **SARIMAX** captures seasonality and leverages external variables.
- **Prophet** provides robust decomposition of trend and seasonality.

The hybrid approach enables the framework to adapt to different data characteristics, achieving better overall prediction performance. For example, SARIMAX improves accuracy when seasonality dominates, whereas XGBoost captures complex interactions in data with dynamic patterns.

### B. Future Work

To further improve drug inventory predictions, the following directions are proposed for future research:

- **Hybrid Framework Expansion**: Extend the hybrid framework by integrating additional models, such as LightGBM, Transformer-based architectures, or ensemble methods to capture both short-term fluctuations and long-term trends.
- **External Variable Enrichment**: Explore additional exogenous variables such as patient inflow, epidemic outbreak data, seasonal illnesses (e.g., flu seasons), or hospital-specific factors to improve prediction accuracy.
- **Automated Model Selection**: Implement automated hyperparameter tuning and model selection techniques (e.g., Bayesian optimization) to improve forecasting performance with minimal manual intervention.

By addressing these areas, the proposed framework can become a more robust and scalable solution for hospital pharmacy inventory management, ensuring continuous patient care and reducing operational inefficiencies.

## V. CONCLUSION

This study evaluated the performance of XGBoost, SARIMAX, and Prophet for drug inventory prediction in hospital pharmacies. The main findings are as follows:

- XGBoost outperformed SARIMAX and Prophet in terms of RMSE and SMAPE for short-term predictions.
- SARIMAX demonstrated better performance when external (exogenous) variables were included and seasonal trends dominated the data.
- Prophet performed well in capturing long-term seasonality and trends but showed limitations in handling highly sparse data.

The hybrid framework, by combining these models, shows promise for improving prediction accuracy and robustness. Future work will focus on integrating real-time data streams and exploring additional hybrid models, such as Transformer-based architectures, to further enhance predictive capabilities.

## VI. REFERENCES

### REFERENCES

- [1] D. Koala, Z. Yahouni, G. Alpan, and Y. Frein, "Factors influencing drug consumption and prediction methods," in *CIGI-Qualita: Conférence Internationale Génie Industriel QUALITA*, Grenoble, France, 2021.
- [2] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.
- [3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
- [4] D. Ferreira, P. Teixeira, and A. Dias, "Exploring the performance of ARIMA and LSTM in time series forecasting: A comparative study," *International Journal of Computer Science and Applications*, vol. 15, no. 2, pp. 20–34, 2018.
- [5] J. Meng, Q. Zhang, and X. Li, "Comparative analysis of Prophet and LSTM models in drug sales forecasting," *Journal of Physics: Conference Series*, vol. 1910, no. 1, p. 012059, 2021.
- [6] W. Xu, Y. Wang, and J. Zhao, "A hybrid modelling method for time series forecasting based on a linear regression model and deep learning," *Applied Intelligence*, vol. 49, no. 7, pp. 2875–2888, 2019.
- [7] R. Siddiqui, A. Khan, and M. Ahmed, "A Hybrid Demand Forecasting Model for Greater Forecasting Accuracy: The Case of the Pharmaceutical Industry," *Supply Chain Forum: An International Journal*, vol. 22, no. 3, pp. 1–13, 2021.
- [8] R. Rathipriya, M. Saranya, and K. Ramkumar, "Demand Forecasting Model for Time-Series Pharmaceutical Data Using Neural Networks," *Neural Computing and Applications*, vol. 35, pp. 1945–1957, 2022.

## APPENDIX

The following pseudocode outlines the sample selection process used to ensure the quality and relevance of the dataset for modeling tasks:

---

### Algorithm 1 Sample Selection Algorithm

---

**Require:** Cleaned data  $df$ , configuration thresholds  $config$   
**Ensure:** Filtered dataset  $final\_df$

```

1:  $final\_df \leftarrow \emptyset$ 
2: for each unique combination of drug name and manufacturer  $(d, m)$  in  $df$  do
3:    $group\_data \leftarrow$  subset of  $df$  for  $(d, m)$ 
4:   if length of  $group\_data$  <  $config.min\_months$  or sum of consumption = 0 then
5:     Skip group ▷ Insufficient or sparse data
6:   end if
7:    $non\_zero\_ratio \leftarrow$  proportion of non-zero consumption in  $group\_data$ 
8:   if  $non\_zero\_ratio$  <  $config.sparsity\_threshold$  then
9:     Skip group ▷ Data too sparse
10:  end if
11:   $acf\_values \leftarrow$  autocorrelation function of consumption in  $group\_data$ 
12:  if  $\max(acf\_values[1 :])$  <  $config.min\_acf\_threshold$  then
13:    Skip group ▷ Insufficient autocorrelation
14:  end if
15:  if no start date  $\geq config.min\_start\_date$  in  $group\_data$  then
16:    Skip group ▷ No recent data
17:  end if
18:   $variance \leftarrow$  variance of consumption in  $group\_data$ 
19:  if  $variance$  <  $config.min\_variance\_threshold$  then
20:    Skip group ▷ Variance too low
21:  end if
22:   $missing\_ratio \leftarrow$  maximum missing ratio for features in  $group\_data$ 
23:  if  $missing\_ratio$  >  $config.max\_missing\_ratio$  then
24:    Skip group ▷ Feature missing data too high
25:  end if
26:   $skewness \leftarrow$  skewness of consumption in  $group\_data$ 
27:  if  $|skewness|$  >  $config.max\_skewness$  then
28:    Skip group ▷ Target variable too skewed
29:  end if
30:   $correlation \leftarrow$  correlation of consumption with lagged features in  $group\_data$ 
31:  if  $|correlation|$  <  $config.min\_correlation$  then
32:    Skip group ▷ Insufficient correlation with features
33:  end if
34:   $final\_df \leftarrow final\_df \cup group\_data$ 
35: end for
36: return  $final\_df$ 

```

---